

Guide S47 - Process Modelling Guidelines

Process modelling is an important part of business design, business redesign, or business re-engineering. Initial guidelines for process modelling have been developed and should be considered for each business design project in the Ministry of Forests and Range.

Process modelling, by itself, is not BPR (Business Process Reengineering) or TQM (Total Quality Management). Process modelling is merely a tool that provides a means of communicating complex business functions in a form more easily understandable by people. Modelling provides for the formalization of processes which in turn allows the business to operate in a standardized manner. Effective design of business processes allows individuals to work together more efficiently. Modelling of lower level (elementary) processes provides consensus on business rules.

Process models are an aid to the understanding of the nature in which processes dynamically work with data. From a design perspective, modelling provides explicit guidelines for modularity, re-usability, flexibility and integrity.

NOTE: These are DRAFT standards

Guide S47 - Process Modelling Guidelines is intended to provide an initial understanding of process modelling as well as identifying the minimum deliverables expected from business design projects. The guide's table of contents includes:

Introduction

About This Guide

The purpose of this guide is to provide guidelines for process modelling deliverables for business design projects within the Ministry of Forests and Range. Related documents include [Guide S7, Data Modelling Guide](#).

The audience for this guide is Ministry of Forests and Range staff and consultants involved with business design (or business redesign, business re-engineering) projects for the ministry.

The Data Administration Section of the Information Management Group is the owner/custodian of this guide.

Creation date: July 15, 1996

Overview - Guidelines for Process Modelling

Process modelling is an important part of business design, business redesign, or business re-engineering. This document describes the process model guidelines that should be considered for each business design project in the Ministry of Forests and Range.

Process modelling, by itself, is not BPR (Business Process Reengineering) or TQM (Total Quality Management). Process modelling is merely a tool that provides a means of communicating complex business functions in a form more easily understandable by people. Modelling provides for the formalization of processes which in turn allows the business to operate in a standardized manner. Effective design of business processes allows individuals to work together more efficiently. Modelling of lower level (elementary) processes provides consensus on business rules. Process models are an aid to the understanding of the nature in which processes dynamically work with data. From a design perspective, modelling provides explicit guidelines for modularity, re-usability, flexibility and integrity.

Terminology

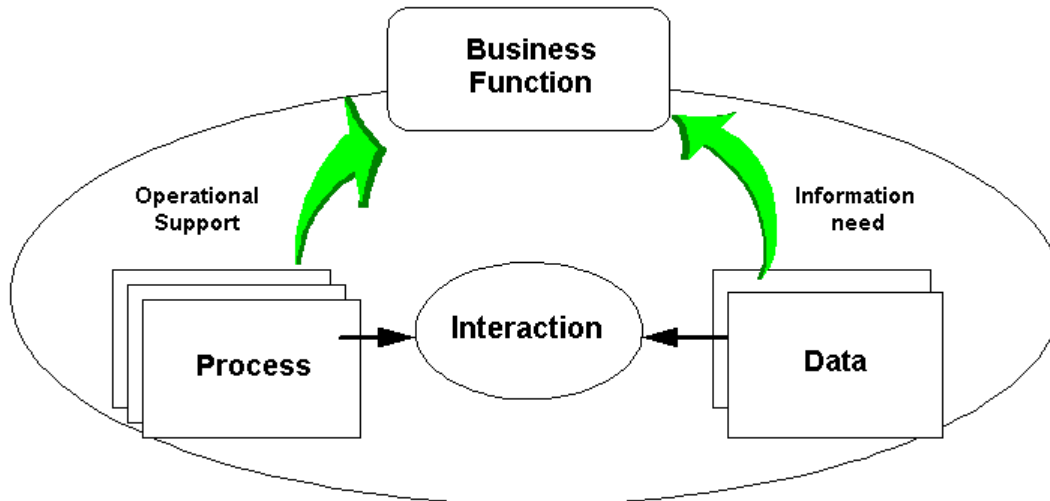
Function:	A group of activities (processes) that together completely support one component of the business.
Process:	A series of repetitive actions with a definable start and end, and well-defined in terms of input, transformation and output of values for entities, relationships and attributes.
Elementary Process:	The smallest unit of activity of meaning to the business and which, when complete, leaves the data in a consistent state. (i.e. An inconsistent state, for example, would occur in the process of a transfer of an employee from one payroll to another payroll, where the employee is only deleted from the former payroll but not added to the new payroll...resulting in no paycheck for the employee.)
Event :	Something that happens outside a process to which the business area must react in a preplanned way. (Sometimes referred to as a trigger or input).
Result :	The planned and expected effect of a business process.
Procedure:	A set of steps for implementing a process using a given technique. For process modelling purposes, procedures are at too low a level to be included.

Contact [Bill Holland](#) for more information.

Process Design

Driven by factors such as the need to increase productivity, operate on reduced budgets, compete in a global economy, and technological innovation, industry has focused on process design improvements to meet these challenges. Business management methods such as TQM or continuous process improvement, or BPR incorporate different techniques to achieve gains in productivity, quality, service etc. Significant results can be achieved from successful projects (e.g. some BPR projects have realized productivity gains of 40-60%, or direct cost reductions of 25 to 60%. Incremental or TQM like projects often realize improvement in the 5 to 15 % range; *Information Systems Management, Summer 1996, Reinventing Through Reengineering. M. A. Mischie & W. Bennis*)

Basic to business design (processing modelling) is the informational component and the interaction between process and data. The process component specifies what processes are to be performed to carry out the business function using the information from the data component. The data component (i.e. documented by a data model) specifies the informational needs of the business. Together, the interaction provides a context in which clarity, completeness and quality can be analyzed, leading to an effective design which meets business objectives.



Processes

Business processes are a set of logically related tasks performed to achieve a defined business result. A set of processes acting on data forms a business system - the method in which an organization carries out its mission. Processes have the following important characteristics:

- Processes are business focused (i.e. "WHAT"). Processes define "what" work needs to be accomplished, not who does the work, or how work will be done. Processes are independent of both technology and organizational structure. They produce clearly defined results.
- Processes decompose into other processes or elementary processes (See later section on process decomposition). Processes have clear boundaries with respect to other processes and are carried out as a single unit of work.
- Processes create or transform information in some way (eg. A process named 'Produce reports' is not a process since it does not transform or create data, it only formats data on a printed page) and leave the information in a stable or complete state. Processes can be coupled to the data model via "views" which identify the inputs to and outputs from the process and upon which the process is dependent.
- Processes may be event-driven. (See later section for discussion and examples of events).
- Processes can be measurable (optional). For example, the measures for the process 'Measure Tree' might be calculated in terms of the length of time it takes to measure a tree, or associated costs, or the number of times trees are measured. It is not the measurement of the tree itself (which is the data the process is acting upon). Process measurement is fundamental to TQM and

Continuous Process Improvement concepts and requires a significant amount of effort by a business area to undertake.

- Processes are typically named in action-object form, for example, 'Receive Order', 'Invoice Customer'.

Events

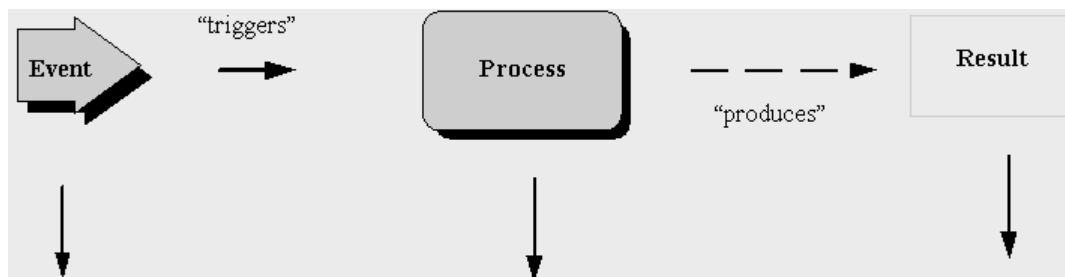
When reasons for the execution of a process are vital or important, typically executed due to the arrival of some information or the passing of some point in time, they are identified as events. An event causes work and is normally thought of as the 'trigger' for a process. Unless event analysis or state transition analysis is to be undertaken, only major events should appear on the process model. However, for each event on a model there must be a corresponding elementary process.

There are two types of events:

- **External** - arrival of information to the process. i.e. customer order arrives; i.e. *fire reported*.
- **Temporal** - point in time; i.e. *fiscal year end, end of pay period*.

In a few cases, there are events that do not have an elementary process since they do not keep or update any data (i.e. public send in coupon, business sends out brochure. If no record is kept of who remitted the coupon, or the number of coupons received or brochures issued, then no work is done - no elementary process).

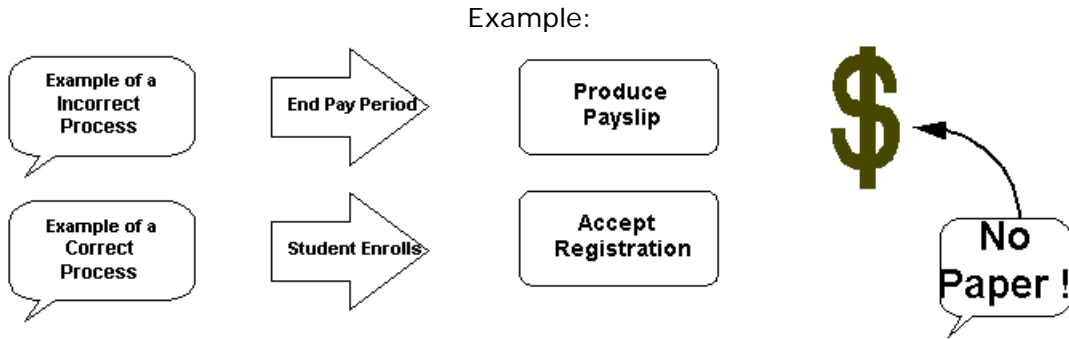
Process Structure



Event: An external activity to the process which needs to be acted on in a preplanned manner.

Process: A series of repeatable well defined actions performing work on data.

Result: Planned and expected effect of a business process. Data is in a consistent state. (Note : results are not identified on a process model.)

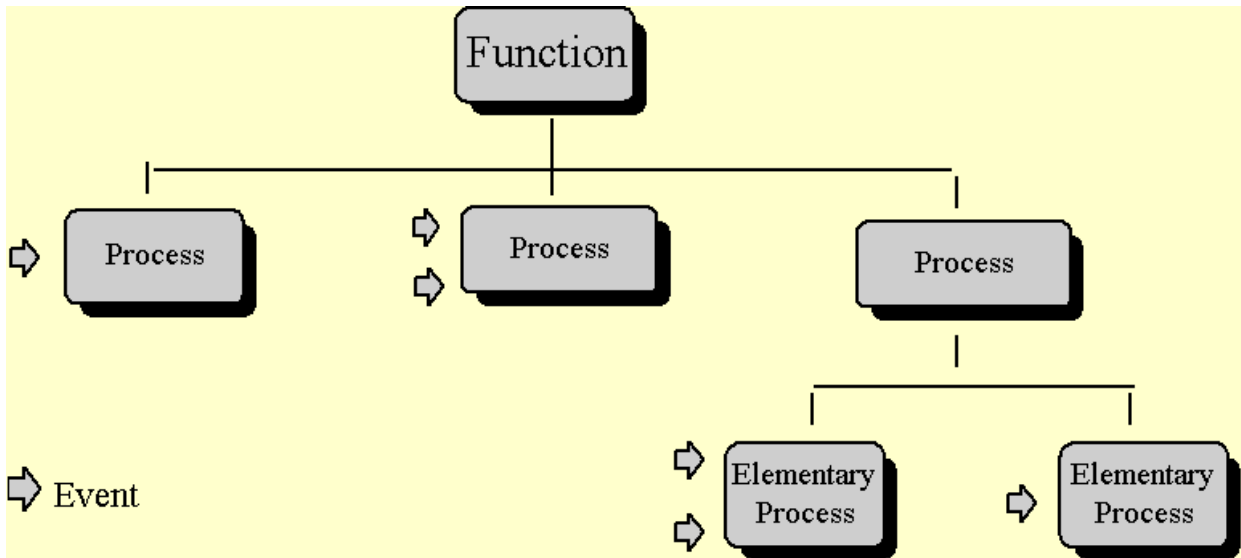


In the first example , the model shows money as the result of a process called Produce Payslip. This process is incorrect. Paper (ie. reports or even the green stuff) is not the result. The process should be named 'Transfer Funds' or 'Update Accounts' and show no paper as a result, similar to second example. The result of the corrected process is that money is transferred and account records updated. The corrected process is shown below.



Process Decomposition

Complex business functions and processes can be broken down into simpler activities. This top-down technique is referred to as *functional decomposition* or *process decomposition*. This technique allows for greater precision in understanding and verifying business objectives.



----- STOP -----

-

The level of decomposition and the rigor applied to the analysis depends on the purpose and scope of the project. Not all projects require analysis to the elementary process level. Projects concerned with information systems development may have more detail than business improvement projects. However, it is important to identify the appropriate point to stop. A level of decomposition that is coherent and has meaning to business must be determined.

The further the decomposition, the more difficult it is to prove uniqueness of lower level processes. Good quality, non redundant process diagrams are not very "deep". In process modelling, too much quantity means too little quality.

Process Dependencies

Business processes do not exist in isolation. Processes require information from other processes, and they in turn provide information to other processes. Dependencies (or associations) between processes occurs when an activity requires information from another activity. Process dependencies are related to the entities and attributes required by the business area. The importance of analyzing and modelling dependencies is to provide further understanding of the interaction between process and data. Process dependency diagrams coupled with the logical data model allow for validation of the interaction between process and data (i.e. does the process need additional information not on the data model, do entity relationships support the process associations). An example of a process dependency diagram and logical data model appear in Appendix A & B respectively.

Elementary Process Validation

The level of process having enough detail so that it can be defined accurately in terms of Entities is called an elementary process. The identification of an elementary process is somewhat subjective, often depending on what the business area considers a single unit of work. Defining elementary processes often requires some consideration of procedures or how processes are to be implemented. This consideration can result in defining processes that are below the elementary process level.

The following tests can be used to assist with validating elementary processes :

Test	Result	Example
Does the process require one and only one event at one time to cause it to occur ?	A business process can have more than one event, but not simultaneously. If more that one event is required at the same time, the process must be split.	A process called Register Cut includes responding to an environmental report. The report arrives separately from the main event. Therefore, Register Cut is not an elementary process.
Can the process be	If not, the process	A process called Cancel Contract

performed without interruption in one place at one time ?	needs to be split	requires recording contract cancellation information, notifying management, and receiving approval from management. Therefore, Cancel Contract is not elementary.
Does the process handle all possible values of data in the incoming flow ?	If not, the process is below the elementary level.	An elementary process Register TFL must be able to recognize that the request is from an unknown client (and reject that request).
Does the process leave the data in a consistent state?	If not, the process is below the elementary level.	Check mandatory relationships on the data model. Mandatory relationships must be formed within the process (or the data model must be changed). e.g. An elementary process Setup Account cannot occur without Client information being present.
Does the process either produce an information flow for an external object or modify an entity, or both ?	If not, it is probably not a process.	The only exception is a process that monitors or controls, where the responses are not explicitly shown (but exist).
Does the process describe: <ul style="list-style-type: none"> • document handling? • conditional situations? • <i>how</i> something is done? 	If so, decomposition has gone too far.	The processes: <ul style="list-style-type: none"> • Print Summary End Report • Amount Over \$500 • Make duplicate copy are not elementary processes.

Rules and Conventions

- Decompose all functions and processes into at least two subordinates down to the elementary process level.
- Decompose a function into either two or more functions or two or more processes.
- Decompose processes into two or more processes.
- Do not mix functions and processes at the same level.
- Most functions or processes do not decompose into more than 7 subordinates.
- Processes appearing more than once on different branches in the same diagram are *reused* processes.
- Elementary processes are the lowest level in the hierarchy and may not be decomposed.

- Each process should convey a single purpose.

Naming Processes

- Use an active verb and object. The object should be singular, as it represents one execution.
- Use more than one object for clarity when required. Object names should normally be an Entity type name or Attribute name. (e.g. Check Stock and Order Parts Quantity).
- Use precise verbs. Verbs such as manage, administer, process, etc. do not convey the precision required for process names. However, they can be used at the function level.
- Use meaningful names that focus on "WHAT", not "how", "when", or "why".
- Avoid jargon and cryptic names. Use words that are familiar to the business area.

Defining Processes

Define the process, do NOT describe it. Define the process in terms of "what" it does to the Entity Relationship Model. For example, a *poor* process definition would be:

Update Account:

This process is an on-line function run on a daily basis at several offices. It requires data entry and performs cost calculations. Data must be able to be revised. The outcome is the individual's account is current.

A *better* definition for the same process would be:

Apply Daily Transaction:

Daily financial transactions are required to be applied to the customer's account by recording the transaction type, amount, and date for each transaction occurrence. The result is an accurate and current customer account balance.

The definition must not merely reiterate the process name. The definition must add value to the process name. Subordinate or sub-processes collectively define the parent process. However, they should not repeat the definition of the parent. Definitions should be as short as possible without sacrificing rigor.

Naming Events

- Use an object and active verb. The object should be singular, as it represents one execution.
- Use an Entity name or attribute name for the object in the process name, i.e. Client Order Arrives.

Process Modelling Deliverables

Process Modelling projects should include deliverables such as a process diagram, a process definition, and a dependency diagram. The process model will be produced co-dependent with a completed logical data model. Conceptual or high level process modelling projects may only require the logical data model to be developed to the entity and primary identifier level.

While the Ministry of Forests and Range has standardized on the use of Oracle Designer for data modelling, a standard tool for process modelling has not been selected. The examples of deliverables that appear in [Appendix A](#) have been produced with IEF, the Texas Instruments CASE tool product. IEF allows for the integration of process models, logical data models, and its data dictionary. (**Note:** IEF is no longer used in the ministry.)

Process Model Validation

Roles & responsibilities:

- Participants at a model review should include representation from the Application Custodian, business areas which have responsibility for the identified business processes, and a Data Administration staff person.
- The primary reason for process modelling is more complete data integrity.

Specific procedures for validating process models have not been developed at this time.

Data quality is dependent on process maturity. Integrateable data is not achievable from ad hoc processes.

Appendix A - Example of Process Model Deliverables

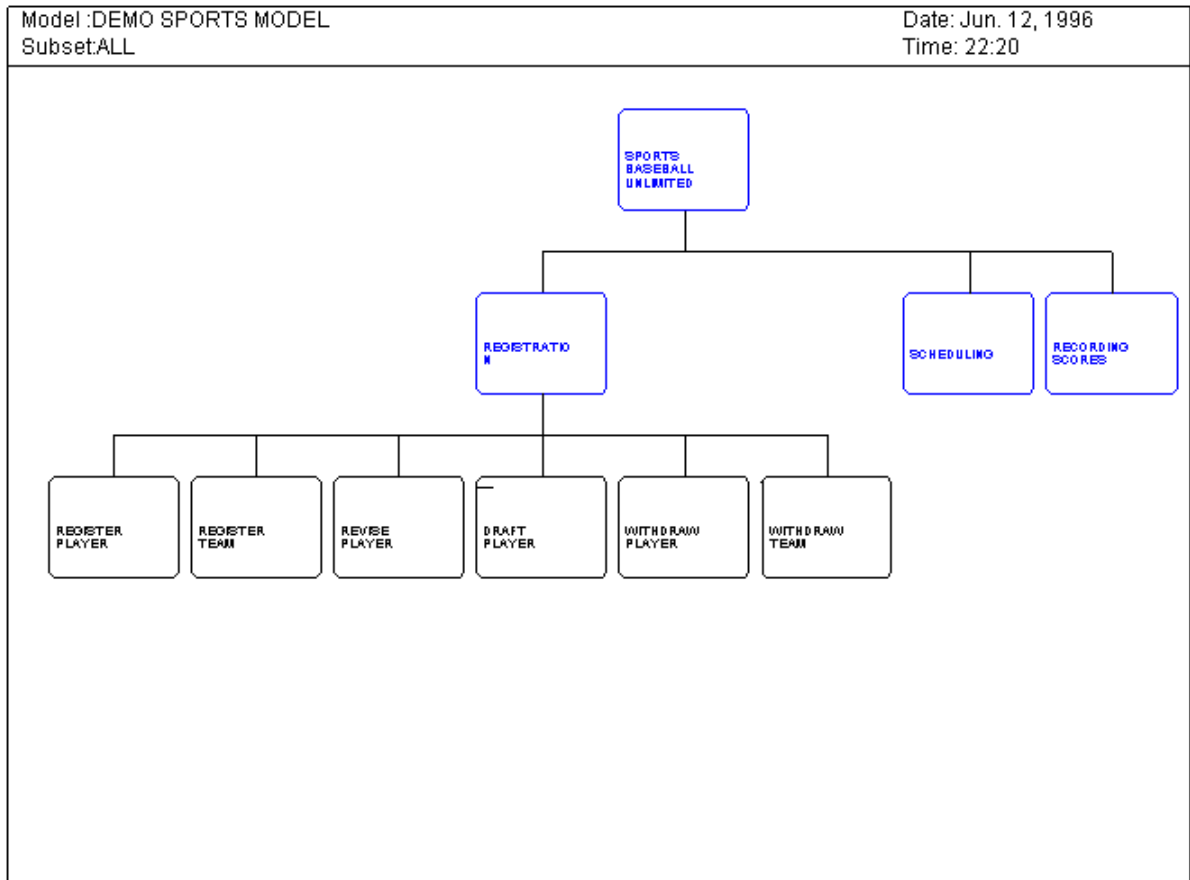
A simple "sports" model was chosen to be modelled as an example for this document deliberately because it is unrelated to any business area in the Ministry of Forests and Range. The purpose of this example is to illustrate and describe minimum process model deliverables and promote an understanding of process modelling, not to provide an MOFR example that has been analyzed and validated by the appropriate business area. The models and reports were produced using IEF.

1. Activity Hierarchy Diagram (Process Diagram)

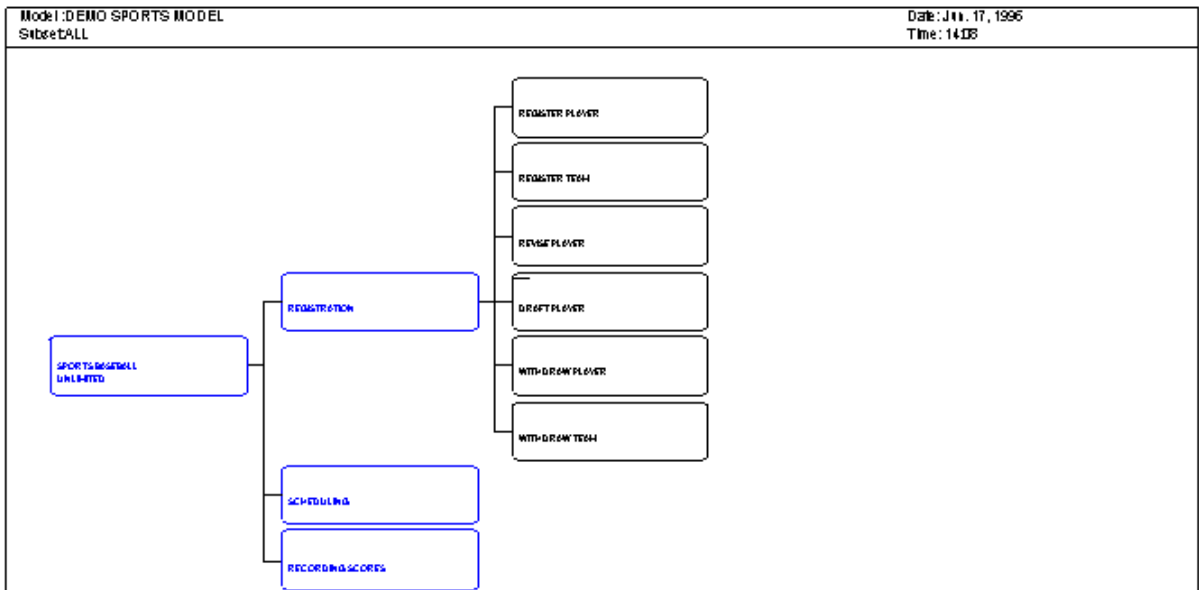
An Activity Hierarchy Diagram shows the functional / process decomposition of a business area. The example, shown below both vertically and horizontally, contains:

- 1 high level function: Sports Baseball Unlimited
- 3 lower level functions: Registration, Scheduling, Recording Scores
- 1 process: Draft Player
- 5 elementary processes: Register Player, Register Team, Revise Player, Withdraw Player, Withdraw Team (these processes were obtained by decomposing the function *Registration*).

Horizontal example:



Vertical example:



3. Activity Definition (Process Definition)

Activity definition details the information required to have complete in the model. It lists all functions and processes in the model by name with the associated definition (description). Each activity is identified by type (i.e. elementary, process), whether the process is repetitive (i.e. Draft Player), and whether a process will be manual or automated (online, batch). All elementary processes in this example are suggested for online implementation with the exception of the process "Draft Player" which will be implemented manually.

Activity Definition Report (example)

Model : DEMO SPORTS MODEL

Subset: (complete model); Jun. 17, 1996 15:01

Name: **REGISTER_PLAYER**

Description: Accept each approved player application prior to the start of the season. The result is a list of players available for team selection.

Type: Elementary process, Not Repetitive

Online implementation suggested

Subordinate of: REGISTRATION

Name: **REGISTER_TEAM**

Description: Accept each approved sponsor by registering team name and team coach prior to the season commencing. The result is a list of teams eligible to select baseball players.

Type: Elementary process, Not Repetitive

Online implementation suggested

Subordinate of: REGISTRATION

Name: **REVISE_PLAYER**

Description: Revise player registry when an individual changes position or jersey number. The result is an accurate list of players and their respective position and number.

Type: Elementary process, Not Repetitive

Online implementation suggested

Subordinate of: REGISTRATION

Name: **WITHDRAW_PLAYER**

Description: Discard player information from player registry when notified of a player's withdrawal. The result is an accurate list of players in the organization.

Type: Elementary process, Not Repetitive

Online implementation suggested

Subordinate of: REGISTRATION

Name: **WITHDRAW_TEAM**

Description: When notified of a sponsorship withdrawal, discard team name from organization. The result is a list of players available for redrafting.

Type: Elementary process, Not Repetitive

Online implementation suggested

Subordinate of: REGISTRATION

Name: **RECORDING_SCORES**

Description: Maintain game results for all games played or defaulted by each team in the organization.

Type: Function

Subordinate of: SPORTS_BASEBALL_UNLIMITED

Name: **REGISTRATION**

Description: Manage the registration of players and teams for the organization. Maintain accurate player and team information , including player movement, team assignment, as well as sponsor or player withdrawal.

Type: Function

Subordinate of: SPORTS_BASEBALL_UNLIMITED

Subordinates: REGISTER_PLAYER

REGISTER_TEAM

REVISE_PLAYER

DRAFT_PLAYER

WITHDRAW_PLAYER

WITHDRAW_TEAM

Name: **SCHEDULING**

Description: Manage the scheduling of games to be played by each team. Over the course of the season, each registered team will play an equal number of games.

Type: Function

Subordinate of: SPORTS_BASEBALL_UNLIMITED

Name: **SPORTS_BASEBALL_UNLIMITED**

Description:

Type: Function

Subordinates: REGISTRATION

SCHEDULING

RECORDING_SCORES

Name: **DRAFT_PLAYER**

Description: All registered and non-drafted players in the organization are selected to play for a registered team. Each team chooses one player until all players are selected and given a jersey number. The result is all players are assigned to a team.

Type: Process, Repetitive

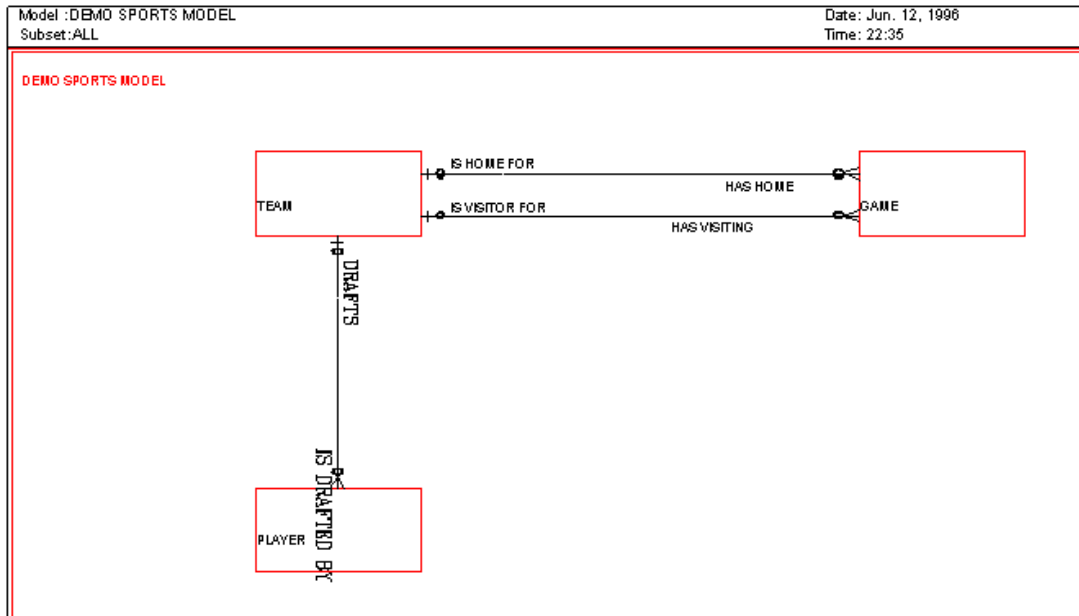
Manual implementation suggested

Subordinate of: REGISTRATION

Appendix B -- Example of Logical Data Model

A logical data model of the "sports" area has been included in this document as a reminder that process models and data models are co-dependent. Utilizing both models in business design analysis will provide a higher degree of data integrity and ensure that business informational needs are identified.

Entity Relationship Diagram



Entity Definition

Model : DEMO SPORTS MODEL

Subset: (complete model)

Entity: **GAME**

Description: Scheduled list of games played or to be played for the current baseball season.

Subject area: DEMO_SPORTS_MODEL

Properties: Min Occ: 0 Avg Occ: 0

Max Occ: 0 Growth Rate: 0% per year

Attributes: DATE, TIME, FIELD, SCORE

Relationships: Sometimes (0%) HAS_VISITING one TEAM, cannot transfer.

Sometimes (0%) HAS_HOME one TEAM, cannot transfer.

Identifiers: DATE, TIME, FIELD

Entity: **PLAYER**

Description: An individual registered with the organization to play baseball.

Subject area: DEMO_SPORTS_MODEL

Properties: Min Occ: 0 Avg Occ: 0

Max Occ: 0 Growth Rate: 0% per year

Attributes: NAME, POSITION, PHONE_NUMBER, JERSEY_NUMBER

Relationships: Sometimes (0%) IS_DRAFTED_BY one TEAM, cannot transfer.

Identifiers: NAME

Entity: **TEAM**

Description: Name of a registered team sponsored with the organization to play baseball.

Subject area: DEMO_SPORTS_MODEL

Properties: Min Occ: 0 Avg Occ: 0

Max Occ: 0 Growth Rate: 0% per year

Attributes: NAME, COACH, COACH_PHONE

Relationships: Sometimes (0%) IS_VISITOR_FOR many GAME; Cardinality Min: 1 (est) Max: 1 (est) Avg: 1, cannot transfer.

Sometimes (0%) IS_HOME_FOR many GAME; Cardinality Min: 1 (est) Max: 1 (est) Avg: 1, cannot transfer.

Sometimes (0%) DRAFTS many PLAYER

Cardinality Min: 1 (est) Max: 1 (est) Avg: 1, cannot transfer.

Identifiers: NAME